ISSN 1561-5359. Штучний інтелект, 2016, № 2

УДК 004.023

О.І.Марченко, О.О. Марченко, М.М. Орлова
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Просп. Перемоги, 37, м. Київ, 03056, Україна

КЛАСИФІКАЦІЯ СПОСОБІВ ПОКРАЩЕННЯ ПОШУКУ ПО ДЕРЕВУ МЕТОДОМ МОНТЕ-КАРЛО, ОРІЄНТОВАНИХ НА ОСОБЛИВОСТІ ЦЬОГО МЕТОДУ

O.I.Marchenko, O.O.Marchenko , M.M. Orlova
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
Peremogy ave., 37, c.Kyiv, 03056, Ukraine

CLASSIFICATION OF MONTE-CARLO TREE SEARCH ENHANCEMENT TECHNIQUES ORIENTED TO SPECIFICS OF THE METHOD

У статті, на основі інформації з різних джерел про пошук по дереву методом Монте-Карло (MCTS), пропонується уточнена структура класифікації та перша версія класифікації способів покращення базової реалізації методу MCTS. У цій версії, на даний момент, розглянуті тільки суто теоретичні способи покращення етапів загальної схеми MCTS, які орієнтовані на особливості роботи цього методу. Передбачається, що запропонована класифікація може бути в подальшому розширена і використана для систематизації знань про метод MCTS та виявлення нових можливостей його покращення.
Ключові слова: задачі штучного інтелекту, дерева ігор, пошук в дереві, метод Монте-Карло, MCTS.
In the article basing on information taken from various sources about Monte-Carlo tree search (MCTS) method, the updated structure of classification and the first version of just the classification of improvement techniques of the basic MCTS method implementation are proposed. For the moment, this version of the classification discusses only pure theoretical techniques for improving of steps of the general MCTS schema, which are oriented to specifics of the method. It is supposed that the proposed classification can be used for systematization of knowledge about MCTS method and discovering of new possibilities for its improvement.
Keywords: artificial intelligence tasks, game trees, tree search, Monte-Carlo method, MCTS.

Вступ

Однією з поширених форм подання інформації в задачах штучного інтелекту є форма у вигляді дерев послідовних рішень, а одним з методів, що був запропонований відносно недавно і призначений для виконання швидкого пошуку правильних рішень у дереві інформації, є метод пошуку в дереві з використанням методу Монте-Карло (Monte Carlo Tree Search – MCTS) [1, 2]. MCTS є методом пошуку оптимальних рішень у заданій області за допомогою випадкових значень із заданого простору значень і побудови дерева пошуку за отриманими результатами. Цей метод вже встиг суттєво вплинути на розв'язання задач штучного інтелекту, а особливо - складних ігрових задач.

Постановка проблеми

За час існування методу MCTS було створено багато його варіантів, з метою прискорення процесу пошуку. Перший фундаментальний огляд способів реалізації MCTS був зроблений в 2012 у [2], але, незважаючи на його ґрунтовність, він не є строгою класифікацією. Крім того, після 2012 року з'явились не тільки нові модифікації та покращення існуючих способів, але й принципово нові варіанти реалізації MCTS. Зазначимо, що задача повної класифікації є об'ємною і виходить за рамки однієї статті, тому в даній статті автори зосереджуються на першій частині

класифікації, присвяченій дослідженню способів реалізації MCTS, що орієнтовані на особливості роботи цього методу.

Мета дослідження

Метою даної роботи є систематизація та класифікація інформації про способи реалізації та покращення методу пошуку в дереві методом Монте-Карло (MCTS), які базуються на покращенні основних етапів (кроків) загальної схеми роботи цього методу, враховуючи особливості їх виконання.

Структура класифікації

Як вже зазначалося, в [1] авторами були запропоновані структура та критерії класифікації різних способів реалізації методу MCTS, його покращення та паралелізації. Був розглянутий підхід до класифікації як для суто теоретичних способів покращення та налагодження методу MCTS для конкретних ситуацій, так і для способів пришвидшення роботи програм, що реалізують цей метод, за допомогою паралелізації та апаратних засобів. За час після першої публікації автори суттєво розширили перелік джерел інформації, в результаті вивчення яких загальна структура класифікації була уточнена. Нова структура класифікації верхнього рівня способів реалізації та покращення методу MCTS показана на рис.1.

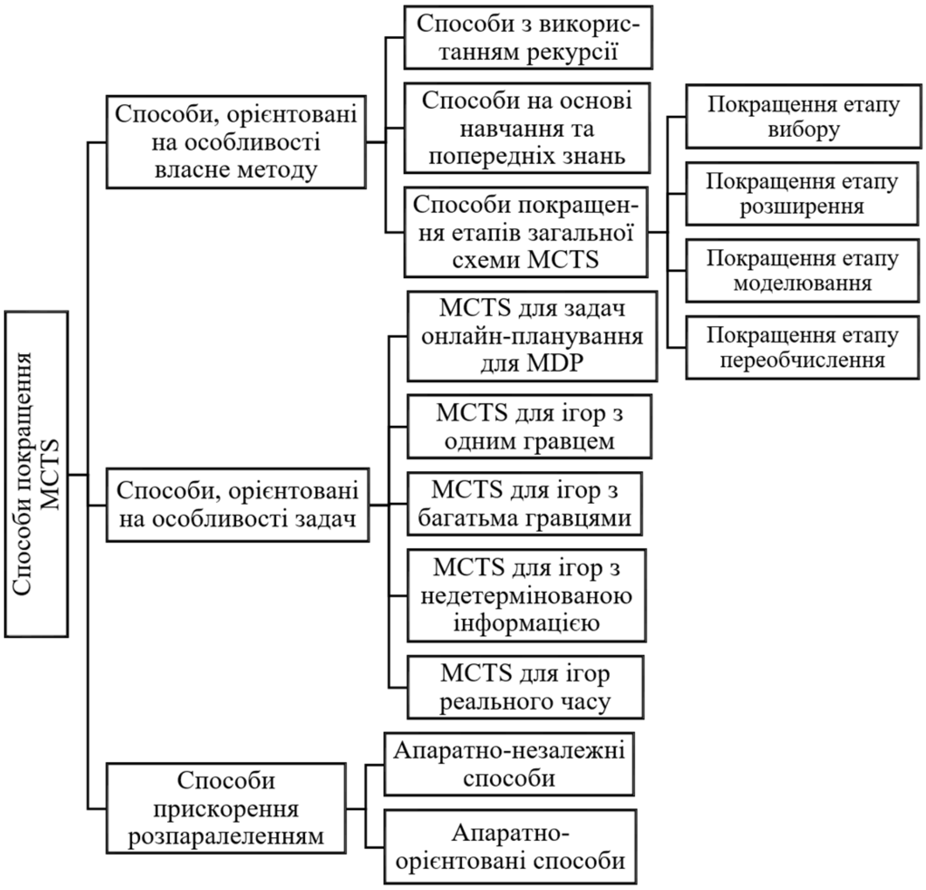


Рис. 1. Структура класифікації способів реалізації та покращення методу MCTS.

На відміну від структури, визначеної в [1], на першому рівні класифікації виділено три групи способів: способи, що орієнтовані на особливості власне методу з теоретичної точки зору [3–15]; способи, що орієнтовані на особливості певних типів задач (як правило, ігор) [16–19]; та способи, що використовують розпаралелення процесу роботи пошуку MCTS [20].

Розглянемо другий рівень класифікації. Структура класифікації групи способів розпаралеленням була запропонована у [1], але власне класифікація за цією групою не є предметом даної статті. Щодо групи способів, орієнтованих на особливості конкретних категорій задач, потрібно зазначити, що способи, які належать до цієї групи, певним чином перетинаються з іншими групами класифікації. Причиною, чому ці способи були виділені у окрему групу, є те, що вони, окрім покращень загального характеру, використовують алгоритмічні ідеї, які можуть бути ефективними тільки для певних категорій задач. Порівняно з [1] була суттєво доопрацьована структура класифікації групи загальних способів покращення MCTS. У цій групі, окрім підгрупи «Способи покращення етапів загальної схеми MCTS», були виділені ще три підгрупи способів, показаних на рис.1, які, на відміну від способів першої підгрупи, орієнтованих тільки на покращення власне етапів схеми MCTS, використовують ще й інші алгоритмічні прийоми загального характеру, такі, як рекурсія або навчання. Саме тому ці способи і були винесені в окремі категорії класифікації.

Класифікація

Як вже відзначалось, усі зазначені вище групи способів неможливо розглянути в рамках однієї статті, і тому в даній роботі розглянута класифікація згідно з визначеними критеріями та структурами тих способів реалізації MCTS, що належать тільки до однієї з груп способів, а саме - «Способи покращення етапів загальної схеми MCTS». Класифікація конкретних реалізацій методу MCTS, що належать підгрупам зазначеної групи способів, показана на рисунках 2, 3, 4 та 5. Щодо деяких інших груп способів будуть також зроблені зауваження, а також наведені джерела інформації про ці способи у переліку літератури до даної статті. Для правильного розуміння рисунків класифікації необхідно зазначити, що блоки класифікації, які відповідають конкретним способам реалізації MCTS (термінальні вершини чи «листки» класифікації), показані на рисунку із затемненням сірим кольором, в той час, як нетермінальні вершини дерева класифікації (абстрактні рівні) показані звичайними блоками на білому фоні. Як відомо, метод пошуку по дереву з використанням методу Монте-Карло (MCTS) складається з чотирьох етапів (кроків): етапу вибору, етапу розширення, етапу моделювання і етапу переобчислення [1, 2]. Відповідно до цього, класифікування способів групи «Способи покращення етапів загальної схеми MCTS» доцільно виконувати саме за критерієм цих етапів. Розглянемо класифікацію способів покращення MCTS згідно такої структури.

Класифікація способів, що покращують етап вибору (політики дерева), показана на рис.2.

Основою групи UCT-орієнтованих способів є рішення так званої задачі багаторуких бандитів (Multi-Armed Bandit – MAB) [2], зокрема, алгоритм UCB (Upper Confidence Bound) визначення верхньої довірчої границі. Цю групу способів можна поділити на різні варіанти реалізації з використанням стандартного алгоритму UCB1 (так званий Vanilla-UCT спосіб) та модифіковані UCT-орієнтовані способи, які мають суттєві відмінності/покращення чистого підходу UCB1. Розглянемо ці модифікації.

Спосіб KL-UCB [3] пропонує покращену UCB-формулу, модифікація якої полягає у використанні дивергенції Кулбека-Лейблера (Kullback-Leibler divergence) для обчислення нижньої границі втрат (regret).

У способі Mi-UCT [4] реалізується покращене обчислення формули UCB, яке полягає у використанні двох відмінних рис: «жадібного» алгоритму вибору наступної вершини дерева (ходу гри) і підтримки лічильника потенційно кращих варіантів вибору замість підтримки множини просто кандидатів вибору.

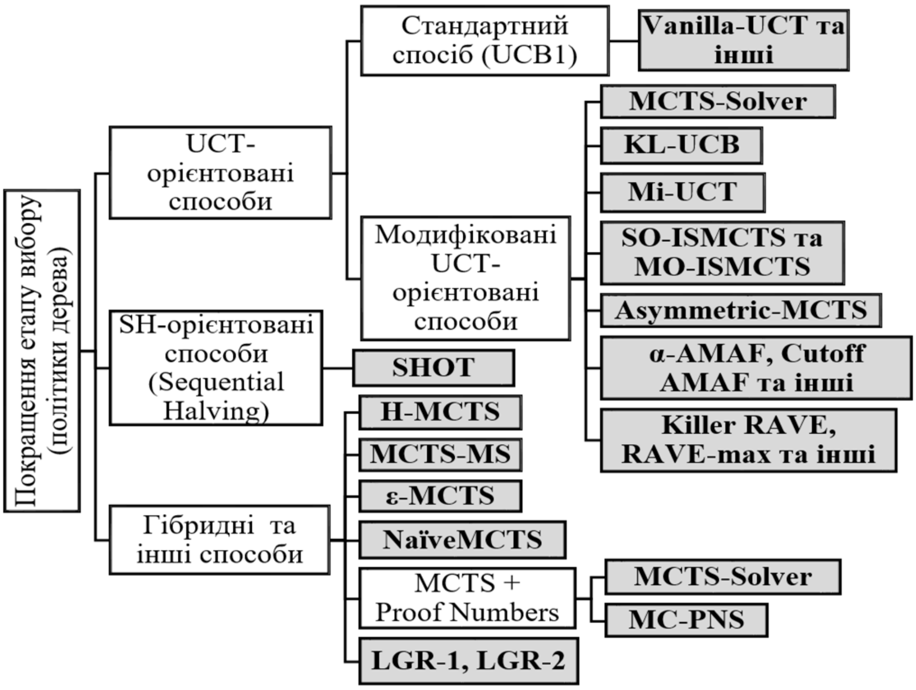


Рис. 2. Класифікація способів покращення етапу вибору (політики дерева).

MCTS-Solver [5] – це спосіб реалізації MCTS, який дозволяє визначати ігро-теоретичну цінність конкретних ходів під час гри. Це досягається завдяки використанню значення ∞ або -∞ додатково до стандартних значень (-1, 0, 1) на етапах вибору та переобчислення.

Способи SO-ISMCTS та MO-ISMCTS [6] дозволяють оперувати різними джерелами прихованої інформації в іграх, оскільки використовують дерева інформаційних множин замість стандартних дерев ігрових станів. Дерево інформаційних множин є фактично перетвореним деревом гри, в якому стани і гілки з однаковою інформацією об’єднуються. Це дозволяє краще аналізувати структуру позицій під час гри.

Asymmetric-MCTS [7] є асиметричним MCTS, який, замість одного алгоритму на етапах вибору та переобчислення, використовує два: алгоритм найменших втрат для max-вершин дерева і стандартний UCB-алгоритм для min-вершин дерева.

Способи групи AMAF (All-Moves-As-First) [2, 8] використовують евристику «всі ходи як перші (All-Moves-As-First)», яка полягає в тому, що під час етапів вибору і переобчислення значення у вершинах дерева оновлюються для всіх ходів, виконаних під час етапу моделювання, а не тільки для одного, як в стандартній схемі MCTS, в результаті чого статистика в вершинах дерева накопичується значно швидше.

Способи групи RAVE (Rapid Action Value Estimates), такі, як Killer RAVE, RAVE-max [2, 9, 10], також використовують евристику AMAF, але обчислюють статистику окремо для кожної вершини дерева гри і комбінують RAVE-оцінку з формулою UCB.

Ефективність UCT-орієнтованих способів досягається завдяки внесенню взаємозв’язаних модифікацій на етапах вибору та переобчислення, бо значення вершин дерева, що використовуються в формулі UCB, оновлюються під час виконання етапу переобчислення. Тому більшість з цих способів показана як на рис.1, так і на рис.5.

Серед способів покращення етапу вибору, окрім вже класичного UCT, у 2015 році був запропонований спосіб SHOT [11], що полягає у використанні принципово іншого підходу на етапі вибору, який називається послідовним половинним діленням

(Sequential Halving – SH). Крім того, було вже запропоновано багато різних гібридних та інших (не-UCT) реалізацій MCTS. Розглянемо основні варіанти таких способів.

Одним із популярних підходів до створення гібридних способів є комбінування UCT з найбільш використаним у задачах штучного інтелекту, до появи MCTS, алгоритмом пошуку Minimax з αβ-відсіканням [13]. Спосіб MCTS-MS (MCTS with Minimax Selection) є представником цього напряму, який використовує таку комбінацію на етапі вибору MCTS, завдяки чому виконує пошук перевірками вершин-нащадків на невелику глибину по дереву і на повну ширину від поточної вершини.

Спосіб ε-MCTS [13] ґрунтується на використанні ε-жадібного (ε-greedy) алгоритму на етапі вибору замість алгоритму багаторукого бандита UCB.

У способі NaïveMCTS [13] реалізується підхід на основі варіанту задачі багаторукого бандита, який називається комбінаторним багаторуким бандитом (Combinatorial Multi-Armed Bandit – CMAB) [13] і полягає в поділі цієї задачі на декілька рівнів, на кожному з яких використовується інший вид формули багаторукого бандита або ε-жадібний алгоритм. Такий підхід дозволяє краще оперувати деревами ігор з великим фактором розгалуження.

H-MCTS [11] є гібридним способом, який об'єднує можливості UCB алгоритму і підходу послідовного половинного ділення SH, що має переваги в частково спостережуваних іграх. У цьому способі алгоритм SH застосовується на вершинах біля кореня дерева гри, в той час, як алгоритм UCB – для всієї іншої частини дерева. Це дозволяє використати стратегію мінімізації простих втрат біля кореня і мінімізацію накопичувальних втрат для інших вершин.

Способи групи MCTS+Proof Numbers [2, 14] є гібридами двох популярних варіантів пошуку по дереву: з використанням методу Монте-Карло (MCTS) та методом доведених чисел (Proof Numbers Search – PNS), що був запропонований раніше. PNS використовує концепцію так званих доведених чисел (proof numbers) на AND/OR деревах. Спосіб MC-PNS використовує оцінювання вершин як у методі MCTS для того, щоб скерувати процес пошуку методом доведених чисел у правильному напрямку і виконувати розширення вершин дерева більш ефективним шляхом. Комбінування ідей MCTS та PNS використовується також у способі MCTS-Solver.

У способах групи Last-Good-Reply (LGR-1, LGR-2) [2, 15] кожен хід у грі розглядається як відповідь на попередній хід і ця інформація зберігається в спеціальній таблиці. Відповідь вважається успішною, якщо вона приводить гравця, що робить цю відповідь, до виграшу. LGR-1 для кожного ходу (включаючи позицію і колір) зберігає тільки одну останню успішну відповідь на цей хід. LGR-2 розширює інформацію способу LGR-1, зберігаючи ланцюжок з двох останніх ходів замість тільки одного попереднього. На основі цієї інформації робиться вибірка наступних ходів.

Перейдемо до способів реалізації MCTS, які використовують покращення на етапі розширення. Такі покращення, як правило, окремо не застосовуються, а використовуються разом з покращеннями на етапах вибору та переобчислення, але далеко не всі модифікації останніх включають одночасне покращення і етап розширення, використовуючи для розширення стандартний спосіб взяття випадкового наступного ходу. Тому на рис.3. зазначені тільки ті з описаних вище способів покращення етапу вибору, які одночасно вносять певні зміни і на етапі розширення.

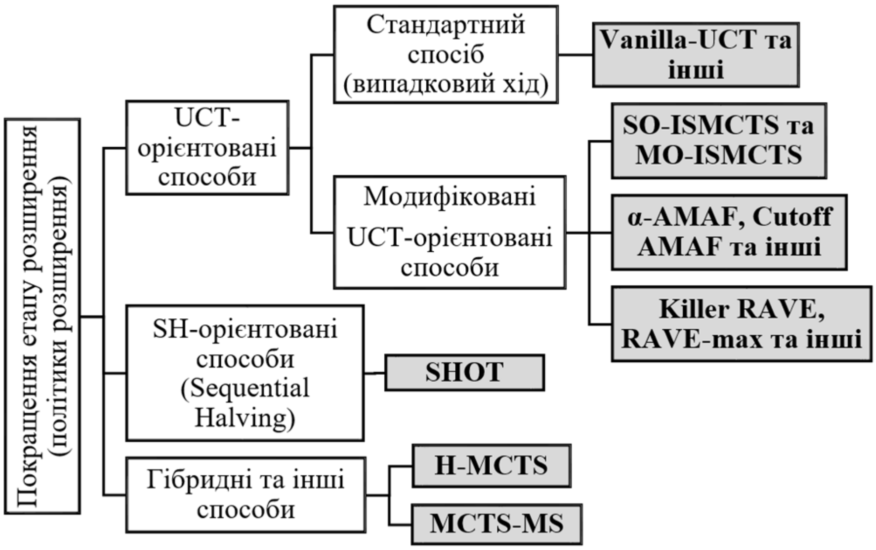


Рис. 3. Класифікація способів покращення етапу розширення (політики розширення).

Розглянемо тепер класифікацію способів покращення MCTS, що вносять певні зміни на етапі моделювання (рис. 4).

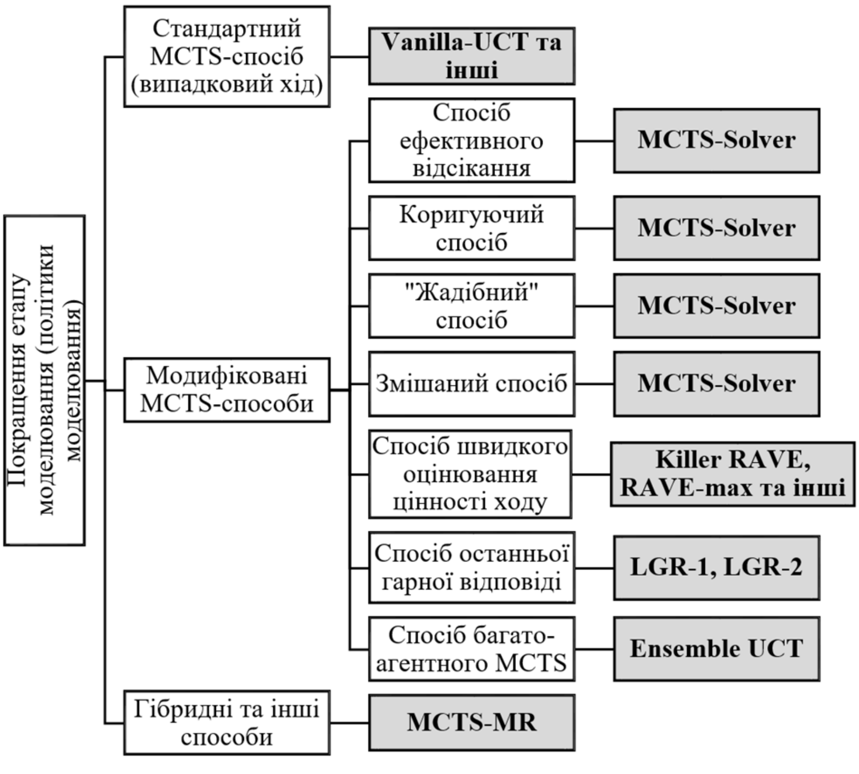


Рис. 4. Класифікація способів покращення етапу моделювання.

Цей етап виконується наступним чином. Починаючи від взятої на етапі розширення вершини, виконується процес моделювання гри до досягнення результату (як правило, виграш/програш) згідно з прийнятою політикою моделювання. Найчастіше береться найпростіша політика моделювання, яка називається політикою за умовчанням (Default Policy) і яка полягає у взятті всіх наступних ходів під час

моделювання випадковим чином. Слід зазначити, що в класифікації етапу моделювання відсутня категорія «UCT-орієнтовані способи», оскільки алгоритм багаторукого бандита і формула UCB прямого відношення до цього етапу не мають.

MCTS-Solver [5] реалізує на вибір відразу чотири модифікації етапу моделювання. Спосіб ефективного (або обчислювального) відсікання (evaluation cut-off) передбачає можливість зупинки (відсікання) процесу моделювання, не обов'язково досягаючи кінцевого результату гри, як при стандартному підході. Наприклад, для гри LOA, якщо термінальна вершина не досягнута, процес моделювання припиняється після моделювання 200 ходів і як результат приймається «нічия». В іншому варіанті, для відсікання моделювань використовується спеціальна оцінювальна функція в процесі моделювання. Якщо її значення стає більшим за заданий верхній пороговий рівень, то результатом моделювання призначається виграш, якщо нижчим за заданий нижній пороговий рівень, то результатом призначається програш. При коригуючому способі використовується оцінювальна функція, що дозволяє змістити вибирання подальших ходів при моделюванні у напрямку мінімізації ризику взяття явно поганого ходу. У «жадібному» способі оцінювальна функція більш прямо застосовується до вибору подальших ходів таким чином, що завжди вибирається хід, що приводить до позиції з найвищим оцінювальним значенням. У змішаному способі використовуються одночасно оцінювальні формули відразу двох способів, коригуючого і «жадібного», при цьому коригуючий спосіб використовується при виборі подальших ходів моделювання до певного порогового значення T, після чого застосовується оцінювальна формула «жадібного» способу.

У способах групи швидкого оцінювання цінності ходу RAVE (Killer RAVE, RAVE-max та інші) [2, 9, 10] етап моделювання виконується відмінним від стандартного (тільки від однієї обраної вершини дерева) підходом, при якому моделювання виконується для кожної з вершин піддерева від обраної вершини (поточного кореня пошуку). Крім того, обчислення цінності ходів також виконується інакше, ніж при стандартному способі UCT.

Способи LR-1, LR-2 з групи останньої гарної відповіді (Last-Good-Reply) [15] на етапі моделювання, при виборі чергового ходу, якщо можливо, використовують саме останню гарну відповідь на попередній хід. У принципі можливі ситуації, коли жодної гарної відповіді ще не зафіксовано або коли збережена відповідь є некоректною для поточної позиції, тому що даний контекст відповіді відрізняється від того, коли ця відповідь була записана у дерево. В обох останніх ситуаціях використовується стандартна політика випадкового моделювання.

У реалізаціях способу багатоагентного MCTS, таких як Ensemble UCT [16], етап моделювання в тому вигляді, як він використовується в класичному варіанті UCT, вважається одноагентним і пропонується використовувати багато агентів, тобто політик моделювання. Різні агенти в цьому випадку отримуються призначенням різних пріоритетів різним евристикам моделювання. Якщо вдається обрати правильну підмножину агентів, то сила гравця збільшується. Однак, знаходження такої підмножини агентів потребує значної інтенсивності обчислень. Спосіб Ensemble UCT використовує підхід, в якому багато UCT-агентів працюють незалежно і статистика, зібрана в коренях цих агентів, об'єднується для отримання остаточного результату. Цей підхід добре відповідає способу кореневої паралелізації [20].

Розглянемо спосіб MCTS-MR (with Minimax Rollout) [12]. Хоча звичайний спосіб взяття випадково ходу при моделювання є достатнім у багатьох ситуаціях, стратегії, що використовують додаткову інформацію при моделюванні, як правило, мають більшу

продуктивність. Виходячи з цього, логічним є застосувати для вибору ходів моделювання пошук Minimax з фіксованою глибиною. Тобто, пошук методом Minimax у межах заданого горизонту знаходить і використовує виграшний хід та відміняє програшний хід. Якщо ж пошук в межах горизонту не знайшов ні виграшу, ні програшу, то повертається випадковий хід.

Перейдемо до розгляду і класифікації способів покращення MCTS на етапі переобчислення, показаної на рис.5.

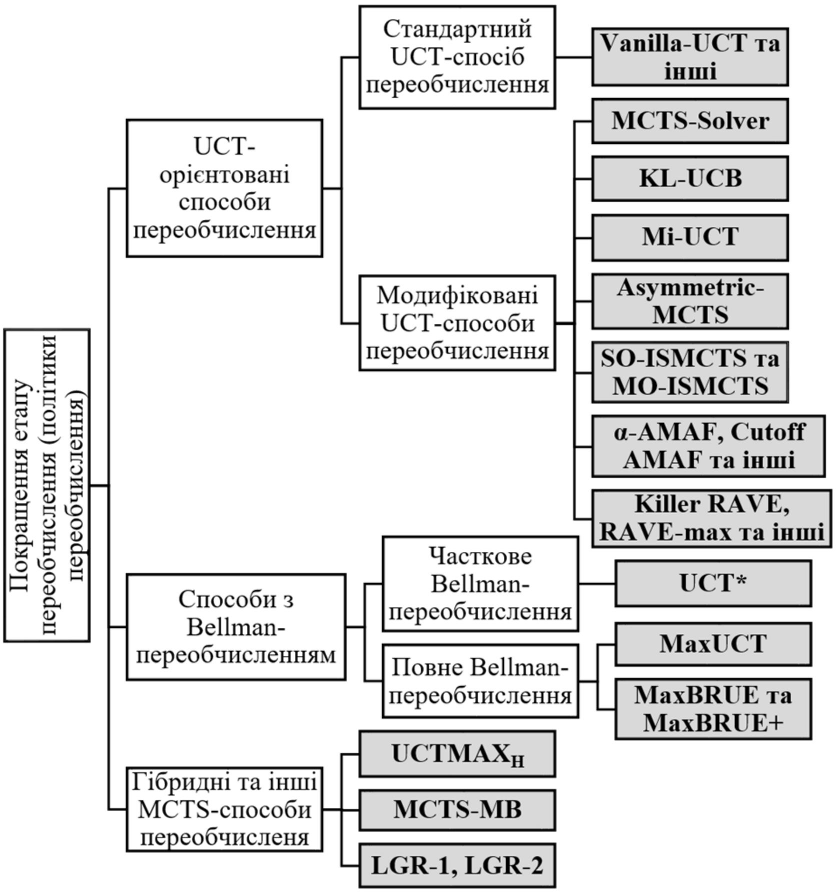


Рис. 5. Класифікація способів покращення етапу переобчислення (політики переобчислення).

При розгляді класифікації способів покращення на етапі вибору було відзначено, що модифікації UCT-орієнтованих способів на етапах вибору та переобчислення взаємозв'язані. Тому розглянемо тільки способи, що не були ще описані раніше.

Способи UCT* [18], MaxUCT, MaxBRUE та MaxBRUE+ [19] використовують Bellman-переобчислення (згідно з формулою Беллмана) для оновлення значень у вершинах дерева замість стандартного для MCTS переобчислення. У той час, як в останніх трьох способах застосовується повне Bellman-переобчислення, то в способі UCT* – часткове (зважене пропорційно до ймовірності в піддереві).

У гібридних способах $UCTMAX_H$ [17] та MCTS-MB [12] етап переобчислення виконується подібно до того, як це робиться в класичному способі Minimax. Нагадаємо, що в оригінальному варіанті UCT, на етапі переобчислення, отримана після багатьох моделювань інформація усереднюється. Така стратегія цілком ефективна, якщо моделювання виконуються випадково. Однак, коли на попередніх етапах вже

використовувались деякі оцінювальні евристики, то переобчислення у стилі Minimax дозволяє отримати набагато кращі результати.

Як раніше зазначалося, у способах групи Last-Good-Reply (LGR-1, LGR-2) [15] кожен хід у грі (за виключенням першого) розглядається як відповідь на попередній хід і ця інформація зберігається в спеціальній таблиці. На етапі переобчислення, в кінці кожного моделювання, кожен виграшний хід зберігається як гарна відповідь на свого попередника, а раніше записана в таблиці інформація перезаписується.

Висновки

У даній роботі були уточнені запропоновані авторами раніше структура та критерії класифікації різних способів реалізації методу MCTS, його покращення та паралелізації.

Після цього, згідно з визначеною структурою, була виконана класифікація інформації про способи реалізації та покращення методу пошуку в дереві методом Монте-Карло, які базуються на покращенні основних етапів (кроків) загальної схеми роботи цього методу, враховуючи особливості їх виконання. Зважаючи на обмежені можливості однієї статті, інші категорії класифікації не були розкриті і є предметом для подальших публікацій.

Передбачається, що як розглянута частина класифікації, так і подальші її розширення, можуть бути використані для систематизації знань про метод MCTS та виявлення нових можливостей його покращення.

Література

1. Марченко О. І. Структура та критерії класифікації способів реалізації та покращення пошуку по дереву методом Монте-Карло / О. І. Марченко, О. О. Марченко, М. М. Орлова. – Комп'ютерно-інтегровані технології: освіта, наука, виробництво, 2015. – № 21. – С. 51–57.
2. Cameron Browne and others. A Survey of Monte Carlo Tree Search Methods // IEEE Trans. on Computational Intelligence and AI in Games. - Vol. 4. - No. 1. - March 2012.
3. Garivier, A., Cappe, A. The KL-UCB algorithm for bounded stochastic bandits and beyond // Proc. 24th Ann. Conf. on Learning Theory (COLT'11), 2011, pp.359-376.
4. Yun-Ching Liu, Yoshimasa Tsuruoka. Adapting Improved Upper Confidence Bounds for Monte-Carlo Tree Search // Proc. 14th Int. Conf., ACG 2015, Leiden, The Netherlands, July 1-3 2015, pp. 53-64.
5. Mark H. M. Winands, Yngvi Björnsson, and Jahn-Takeshi Saito. Monte Carlo Tree Search in Lines of Action // IEEE Trans. Comp. Intell. AI Games. - Vol. 2. - No. 4. - December 2010, pp.239-250.
6. Cowling, P., Powley, E., Whitehouse, D.: Information set Monte Carlo tree search // IEEE Trans. Comp. Intell. AI Games. - Vol. 4. - No. 2. - June 2012, pp.120–143.
7. Yun-Ching Liu, Yoshimasa Tsuruoka. Asymmetric Move Selection Strategies in Monte-Carlo Tree Search: Minimizing the Simple Regret at Max Nodes // Submitted on 8 May 2016 to the 2016 IEEE Comp. Intell. and Games Conference, pp.193-199.
8. D. P. Helmbold and A. Parker-Wood. All-Moves-As-First Heuristics in Monte-Carlo Go // Proc. Int. Conf. Artif. Intell., Las Vegas, Nevada, 2009, pp. 605–610.
9. Sylvain Gelly, Levente Kocsis, Marc Schoenauer, Michèle Sebag, David Silver, Csaba Szepesvári, and Olivier Teytaud. The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions. // Communications of the ACM, - Vol. 55, - No. 3, - March 2012, pp.106-113.
10. Tristan Cazenave. Generalized Rapid Action Value Estimation // Proc. 24th Int. Conf. on Art. Intell., IJCAI 2015, Buenos Aires, Argentina, July 26-27, 2015, pp 754-760.
11. T. Pepels, T. Cazenave, and M.H.M. Winands. Sequential Halving for Partially Observable Games // Proc. 24th Int. Conf. on Art. Intell., IJCAI 2015, Buenos Aires, Argentina, July 26-27, 2015, pp.16-29.
12. Hendrik Baier and Mark H. M. Winands. MCTS-Minimax Hybrids // IEEE Trans. Comp. Intell. AI Games. - Vol. 7. - No. 2. - June 2015, pp. 167-179.
13. S. Ontanon. The combinatorial multi-armed bandit problem and its application to real-time strategy games // Proc. 9th AAAI Conf. Artif. Intell. Interact. Dig. Entertainment 2013, pp.58-64.
14. J.-T. Saito, G. Chaslot, Jos W.H.M. Uiterwijk, and H. J. van den Herik. Monte-Carlo Proof-Number Search for Computer Go // Proc. 5th Int. Conf. Comput. and Games, Turin, Italy, 2006, pp. 50–61.
15. H. Baier and P. D. Drake.The Power of Forgetting: Improving the Last-Good-Reply Policy in Monte Carlo Go Trees// IEEE Trans. Comp. Intell. AI Games. - Vol. 2. - No. 4. - December 2010, pp. 303-309.
16. Fern and P. Lewis. Ensemble Monte-Carlo Planning: An Empirical Study // Proc. 21st Int. Conf. Automat.

Plan. Sched., Freiburg, Germany, 2011, pp. 58–65.
17. R. Ramanujan and B. Selman. Trade-offs in sampling-based adversarial planning // Proc. 21st Int. Conf. Automat. Plan. Sched., Freiburg, Germany, 2011, pp.202-209.
18. Thomas Keller, Malte Helmert. Trial-based Heuristic Tree Search for Finite Horizon MDPs // [Електр. Ресурс]. Режим доступу: http://www.aaai.org/ocs/index.php/ICAPS/ICAPS13/paper/view/6026.
19. Zohar Feldman, Carmel Domshlak. On MABs and Separation of Concerns in Monte-Carlo Planning for MDPs // Proc. 24th Int. Conf. Automat. Plan. Sched., pp.120-127.
20. G. M. J.-B. Chaslot, M. H. M. Winands, and H. J. van den Herik. Parallel Monte-Carlo Tree Search // Proc. Comput. And Games, LNCS 5131, Beijing, China, 2008, pp. 60–71.

Literatura

1. Marchenko O. I. Struktura ta kryteriyi klasyfikatsiyi sposobiv realizatsiyi ta pokrashchennya poshuku po derevu metodom Monte-Karlo / O. I. Marchenko, O. O. Marchenko, M. M. Orlova. – Komp"yuterno-intehrovani tekhnolohiyi: osvita, nauka, vyrobnytstvo, 2015. – № 21. – C. 51–57.
2. Cameron Browne and others. A Survey of Monte Carlo Tree Search Methods // IEEE Trans. on Computational Intelligence and AI in Games. - Vol. 4. - No. 1. - March 2012.
3. Garivier, A., Cappe, A. The KL-UCB algorithm for bounded stochastic bandits and beyond // Proc. 24th Ann. Conf. on Learning Theory (COLT'11), 2011, pp.359-376.
4. Yun-Ching Liu, Yoshimasa Tsuruoka. Adapting Improved Upper Confidence Bounds for Monte-Carlo Tree Search // Proc. 14th Int. Conf., ACG 2015, Leiden, The Netherlands, July 1-3 2015, pp. 53-64.
5. Mark H. M. Winands, Yngvi Björnsson, and Jahn-Takeshi Saito. Monte Carlo Tree Search in Lines of Action // IEEE Trans. Comp. Intell. AI Games. - Vol. 2. - No. 4. - December 2010, pp.239-250.
6. Cowling, P., Powley, E., Whitehouse, D.: Information set Monte Carlo tree search // IEEE Trans. Comp. Intell. AI Games. - Vol. 4. - No. 2. - June 2012, pp.120–143.
7. Yun-Ching Liu, Yoshimasa Tsuruoka. Asymmetric Move Selection Strategies in Monte-Carlo Tree Search: Minimizing the Simple Regret at Max Nodes // Submitted on 8 May 2016 to the 2016 IEEE Comp. Intell. and Games Conference, pp.193-199.
8. D. P. Helmbold and A. Parker-Wood. All-Moves-As-First Heuristics in Monte-Carlo Go // Proc. Int. Conf. Artif. Intell., Las Vegas, Nevada, 2009, pp. 605–610.
9. Sylvain Gelly, Levente Kocsis, Marc Schoenauer, Michèle Sebag, David Silver, Csaba Szepesvári, and Olivier Teytaud. The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions. // Communications of the ACM, - Vol. 55, - No. 3, - March 2012, pp.106-113.
10. Tristan Cazenave. Generalized Rapid Action Value Estimation // Proc. 24th Int. Conf. on Art. Intell., IJCAI 2015, Buenos Aires, Argentina, July 26-27, 2015, pp 754-760.
11. T. Pepels, T. Cazenave, and M.H.M. Winands. Sequential Halving for Partially Observable Games // Proc. 24th Int. Conf. on Art. Intell., IJCAI 2015, Buenos Aires, Argentina, July 26-27, 2015, pp.16-29.
12. Hendrik Baier and Mark H. M. Winands. MCTS-Minimax Hybrids // IEEE Trans. Comp. Intell. AI Games. - Vol. 7. - No. 2. - June 2015, pp. 167-179.
13. S. Ontanon. The combinatorial multi-armed bandit problem and its application to real-time strategy games // Proc. 9th AAAI Conf. Artif. Intell. Interact. Dig. Entertainment 2013, pp.58-64.
14. J.-T. Saito, G. Chaslot, Jos W.H.M. Uiterwijk, and H. J. van den Herik. Monte-Carlo Proof-Number Search for Computer Go // Proc. 5th Int. Conf. Comput. and Games, Turin, Italy, 2006, pp. 50–61.
15. H. Baier and P. D. Drake.The Power of Forgetting: Improving the Last-Good-Reply Policy in Monte Carlo Go Trees// IEEE Trans. Comp. Intell. AI Games. - Vol. 2. - No. 4. - December 2010, pp. 303-309.
16. Fern and P. Lewis. Ensemble Monte-Carlo Planning: An Empirical Study // Proc. 21st Int. Conf. Automat. Plan. Sched., Freiburg, Germany, 2011, pp. 58–65.
17. R. Ramanujan and B. Selman. Trade-offs in sampling-based adversarial planning // Proc. 21st Int. Conf. Automat. Plan. Sched., Freiburg, Germany, 2011, pp.202-209.
18. Thomas Keller, Malte Helmert. Trial-based Heuristic Tree Search for Finite Horizon MDPs // [Електр. Ресурс]. Режим доступу: http://www.aaai.org/ocs/index.php/ICAPS/ICAPS13/paper/view/6026.
19. Zohar Feldman, Carmel Domshlak. On MABs and Separation of Concerns in Monte-Carlo Planning for MDPs // Proc. 24th Int. Conf. Automat. Plan. Sched., pp.120-127.
20. G. M. J.-B. Chaslot, M. H. M. Winands, and H. J. van den Herik. Parallel Monte-Carlo Tree Search // Proc. Comput. And Games, LNCS 5131, Beijing, China, 2008, pp. 60–71.

RESUME

O.I.Marchenko, O.O.Marchenko, M.M. Orlova

Classification of Monte-Carlo tree search enhancement techniques oriented to specifics of the method

In the paper, structure and criteria, which were proposed by the authors earlier for classification of techniques for improvement and parallelization of the Monte-Carlo tree search (MCTS) method implementation, are adjusted. In the updated structure, at the first level of classification three groups of improvement techniques are defined: (1) techniques oriented to specific features of just the method; (2) techniques oriented to domain-specific features of particular types of tasks; and (3) techniques oriented to speedup of MTCS run by parallelizing of searching process.

Then according to the defined structure and basing on large information taken from various sources about MCTS method, specific features of its behavior for various cases of usage, and implementation of this method for many applications, the article proposes a classification of MCTS method implementation and improvement techniques which are based on improvements of main phases (steps) of the general MCTS schema considering specific features of the techniques. Improvement techniques for selection, expansion, simulation and backpropagation phases are discussed separately. In total, more than 20 MCTS implementations and 30 improvement techniques for it were overviewed in the paper.

Taking into account limited size of one article, other classification categories of MCTS improvement techniques (domain-specific and parallelizing improvements) were not discussed now and constitute the subject for some consequent papers. It is supposed that both this part of the classification and its further expanding can be used for systematization of knowledge about MCTS method and discovering new possibilities for its improvement.

Надійшла до редакції 05.10.2016